HAITI II: Human Analytics of Information & Technology Interaction, Part 2

Academic year 2023-2024

Student’s module book

VRIJE UNIVERSITEIT BRUSSEL
somt University of Physiotherapy
Tampere University
Est. 1841 YORK ST JOHN UNIVERSITY

Please think green, consider the environment before printing

Content

Part 1 General information

1.1 Planning group and coordination

…

Role	Name	e-mail	Picture
Main lecturer, slides	Dr. Pekka Kallioniemi	pekka.kallioniemi@tuni.fi	
Assignments and practical coding activities, lectures on visual encodings in VR and real-time visualization	John Mäkelä	john.makela@tuni.fi	
Special lecturer on accessibility, slides	Pauliina Baltzar	pauliina.baltzar@tuni.fi	
Slides	Prof. Markku Turunen	markku.turunen@tuni.fi	

1.2 Introduction

In the last decades the amount of data has exploded, and at the beginning of 2020 the amount of data around the world was estimated to be around 64 zettabytes. At the same time in this fast-paced digital world we must ingest information faster and more efficiently to support decision-making. To support this decision making, the field of information visualization was developed. By transforming abstract information into easily digestible representations, we can support decision making in various fields, including physiotherapy.

This course continues from HAITI I. It is an advanced course into information analytics and visualization, and during the course students will learn how to design and implement complex and interactive visualization dashboards. The focus of this course is on the more advanced techniques of visualization including real-time data visualization, virtual reality/extended reality visualizations, automatic report generation, AI in visualization, accessible information visualization and the theoretical background of visual encodings.

1.3 Content and aim of the module

- Advanced Interaction techniques
- Alternative techniques for information presentation
- Extended reality and artificial intelligence in visualization
- Accessible information visualization
- Theory of visual encodings
- Programming packages: Pandas, Dash

1.4 Educational activities

The students will study according to the study syllabus, learning outcomes, individual and workgroups assignments and lecture outlines. The students will get acquainted on visualizing real-time streaming data and the creation of dashboards with automated report generation. Final assignments will be presented in a seminar at the end of the course.

1.5 Learning outcomes

Upon completion of the course, the student will

1. know the most common information visualization techniques
2. know the most common alternative techniques for information presentation
3. be familiar with advanced information presentation techniques (XR and AI)
4. know how accessible information visualizations can be designed
5. be able to implement and evaluate interactive visualizations for real-world problems
6. know the basics of automatic report generation

1.6 Testing and grading (evaluation)

The grading method for this course is *pass/fail*. To pass the course, students will have to complete a portfolio of all mandatory individual and group assignments and/or give presentations consisting of specific assignments during the course.

Part 2 Lessons

2.1 Lecture 1: Advanced dashboard implementation

Methods

Distance learning (Teams lecture, homework)

Overview

In this lecture, students will learn how to build interactive data visualization dashboards using the Dash framework, a Python-based library that simplifies web-based visualization development. Dash enables users to create dynamic dashboards without needing to write manual code for frontend technologies such as HTML, CSS, or JavaScript. This lecture builds on students' prior experience with Pandas and Plotly during HAITI-I to demonstrate how advanced 2D and 3D visualizations can be incorporated into functional dashboards.
By the end of the lecture, students will understand how Dash allows for real-time updates, user interactions, and data-driven visuals; all managed through Python without the need to handle complex web development components manually.

Key topics

Libraries and Dependencies
Students will learn the importance of importing the appropriate libraries, including Dash, Dash HTML Components, and Pandas, to load and process data. They will also understand the role of Plotly in creating the individual visualizations or figures on their dashboards, enabling both 2D and 3D interactive visualizations.

Application Layout
The lecture will guide students through the structure of a Dash layout, where they will define the appearance and behavior of the dashboard.
Students will be shown how to use Dash components like Div, Graph, and Slider to build the interface, and how to link these components to the data being processed.

Callbacks for Interactivity
One of the key features of Dash is its ability to make elements interactive through callbacks. Students will learn how callbacks enable dashboards to respond to user input, such as updating a graph when a slider is adjusted. The lecture will explain how callbacks are written in Python and how they connect the user interface to the underlying data and visual elements.

Running a Dash Application
Students will be shown how to initialize and run a Dash app on a local server, where they can access the dashboard through a web browser. The lecture will also cover troubleshooting common issues when running Dash applications.

References

https://dash.plotly.com/tutorial

Assignment: Advanced Dash Tutorial

This assignment introduces students to the Dash framework, used for building interactive, web-based data visualization dashboards with Python. It builds upon the skills acquired in previous assignments, where students worked with libraries like Pandas, Plotly, and Vedo for 2D and 3D visualizations.

Objectives
The primary objective is to help students learn how to create dynamic and interactive dashboards using

Met opmerkingen [JM1]: @Pekka Kallioniemi (TAU)
Only you have the slides for the advanced visualizations lecture (latter part of this https://vub.sharepoint.com/sites/FacLK-PG-RhsT/_layouts/15/stream.aspx?id=%2Fsites%2FFacLK%2DPG%2DRhsT%2FGedeelde%20documenten%2F6%2E%20HAITI%2DII%2FRecordings%2F16%5F04%5F2024%5FLecture%2Emp4&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview%2Eeab789a9%2Dfb2f%2D4d7b%2Db579%2D99aabf71aba6)

Dash, without directly writing frontend code like HTML, CSS, or JavaScript. By following the official Dash tutorial and exploring sample applications, students will understand how to structure and run a Dash app and incorporate interactivity into their visualizations.

Assignment Breakdown

Prerequisites

Students are required to install Git for Windows and ensure it's set up for use from both the command line and third-party software.
Dash must be installed via Python's package manager (pip) or from an integrated development environment such as PyCharm.

Dash in 20 Minutes Tutorial

Students will go through a guided tutorial, which introduces the basic structure of a Dash application:

- Importing necessary libraries such as Dash and HTML components.
- Loading and processing data using libraries like Pandas.
- Initializing a Dash application.
- Constructing the layout, which defines how the dashboard will look and behave.
- Running the application on a local server.

The tutorial emphasizes the auto-generation of HTML, CSS, and JavaScript based on the defined layout, effectively managing both the frontend and backend using only Python code.

Running Sample Dashboards

Students will clone a repository containing sample Dash applications. One of the key samples is a clinical analytics dashboard. They will navigate through the repository, install dependencies, and run the provided sample dashboards to see how a fully functioning Dash app works. Students will explore the interaction techniques used in these dashboards, such as dynamic updates and user input handling, to gain insight into how Dash facilitates complex, real-time visualizations with minimal code.

Outcomes

By completing this assignment, students will gain foundational knowledge of Dash, including how to set up a dashboard, handle data, and create interactive visual elements. This knowledge will be crucial for implementing dashboard with Dash.

2.2 Lecture 2: Real-time visualization and data streaming

Methods

Distance learning (Teams lecture, homework, QA session and student presentations for previous assignment)

Introduction

This lecture focuses on the visualization of real-time data, particularly the technical aspects of receiving and processing live sensor data in real-time applications. It explores key concepts such as the distinction between hard and soft real-time systems, the use of networking protocols, and methods for transmitting and receiving data, with a specific emphasis on the User Datagram Protocol (UDP) for live data visualization.

Key Topics

Real-Time Systems

Hard real-time systems require strict deadlines for processing and visualization, where missing a deadline could result in catastrophic failures, common in critical systems such as flight control.
Soft real-time systems, however, aim to meet deadlines but can tolerate occasional delays without

critical consequences. Most real-world applications, including real-time data visualization, are soft real-time systems.

Networking in Real-Time Visualization
Real-time data is typically transmitted from a sensor or device to a computer over a network. The lecture discusses the difference between Local Area Networks (LANs) and Wide Area Networks (WANs): LANs include smaller, wired networks or Bluetooth connections used within close range whereas WANs involve broader networks, like the internet, where more secure protocols may be required.

Communication Protocols
UDP (User Datagram Protocol) is introduced as a lightweight protocol ideal for real-time applications, where packet order and guaranteed delivery are not as important. Examples of sending and receiving data using UDP sockets are discussed.
Bluetooth is another method highlighted for wireless communication over short distances, with a focus on Bluetooth Low Energy (BLE), commonly used for IoT devices and wireless sensors.

UDP Data Transmission
Practical code examples in Python demonstrate how to transmit and receive sensor data using UDP. This method is suitable for applications where low latency is more critical than reliability, such as live data streaming.

Open Sound Control (OSC)
As an alternative to UDP, OSC is discussed for multimedia and sensor applications. It supports various data types and is often used in interactive installations and audiovisual performances.

References
https://docs.python.org/3/library/struct.html

https://github.com/attwad/python-osc
https://pandas.pydata.org/docs/reference/api/pandas.read_sql.html
https://github.com/plotly/dash-sample-apps/tree/main/apps/dash-wind-streaming

Assignment 2: Visualization of data streamed over UDP

Objective
In the accompanying assignment, students are tasked with implementing real-time data visualization using Dash and UDP communication. The Python scripts provided (udp_receiving.py and dash_udp_receiving.py) serve as foundational code for receiving and visualizing live data.

udp_receiving.py Overview
This script demonstrates how to receive data over a UDP connection. Using Python's socket library, it opens a UDP socket and listens for incoming data packets. Each packet contains two floating-point values, which are unpacked and printed to the console. This example illustrates basic UDP data reception in a local network.

dash_udp_receiving.py Overview
Building on the previous script, this file integrates real-time data reception with Dash for visualization. The application creates a UDP server that continuously listens for incoming data packets. It updates a live plot in a Dash dashboard at regular intervals using the dcc.Interval component. The main elements of the script include

- A UDP server that binds to a specified IP and port to receive data.
- A Dash layout with a graph component to visualize the incoming data in real-time.
- A callback function that updates the graph based on the data received from UDP packets. The graph shows the most recent data points, limiting the number displayed to the last 50 values.

Outcomes

Students are required to:

- Understand how UDP data is received and processed.
- Implement a simulation of live data transmission by sending packets in a loop with a short delay.
- Modify or extend the provided Dash program code to visualize the simulated data in real-time.

This assignment focuses on combining real-time data handling techniques with data visualization, giving students practical experience in creating dynamic, live dashboards.

2.3 Lecture 3: Theory of visual encodings

Methods

Distance learning (Teams lecture, QA session and student presentations for previous assignment)

Overview

This lecture focuses on the principles of visual encodings and their effectiveness for representing data, extending from traditional 2D visualizations to more immersive 3D and virtual reality (VR) environments. The concepts presented draw heavily from early research in graphical perception and the ranking of visual encodings, particularly the work of Mackinlay (1986) and Cleveland and McGill (1984). The lecture outlines how various types of visual encodings such as position, length, and angle are more or less effective in conveying information accurately to viewers.

Theory of visual encodings

Effectiveness of Graphical Representations

Visual encodings are ranked based on their effectiveness in terms of accuracy. Mackinlay's research, which is foundational for this lecture, established a hierarchy of visual encoding methods through the lens of human graphical perception. Cleveland and McGill further refined this ranking, identifying how certain encodings are better suited for different types of data:

Position: The most effective encoding for accurately conveying information, especially when data points are aligned on a common scale.
Length, Angle, and Area: Other encodings like length and angle are less effective than position but still useful, particularly when position-based representations are impractical.
Volume, Density: These encodings tend to be less precise, particularly when applied in situations that don't naturally involve 3D data or density measures.

Visual Encoding in Different Data Types

Different data types require different encodings for effective visualization
Interval or Ratio Data: Best represented by position, followed by length or angle.
Ordinal Data: Position remains the most effective, with color saturation and texture providing additional encoding options.
Nominal Data: Position is still useful, but features like color hue, texture, and shape are more commonly employed to distinguish categories.

The lecture stresses the importance of selecting the appropriate visual encoding based on the type of data being represented. Using unsuitable encodings can lead to misinterpretation or a loss of precision.

Extending to 3D and VR Environments

As visualization moves from 2D into 3D or VR environments, the effectiveness of visual encodings can change. Although 3D and VR offer the potential for richer, more interactive data visualizations, they introduce new challenges in terms of perceptual accuracy.

Position in 3D: The introduction of the Z-axis adds depth to visualizations, which can be useful for inherently 3D data (e.g., geological or medical data). However, 3D projections onto 2D displays can distort information, leading to a reduction in accuracy.
Volume and Density in 3D: Volume-based encodings, while intuitive in 3D, often suffer from perceptual distortion when viewed in 2D projections. The lecture recommends avoiding unnecessary 3D elements (such as 3D bar charts), suggesting instead the use of multi-panel 2D figures for complex datasets.

Perception of Depth in VR

The lecture further explains the role of depth perception in VR, referencing research by LaValle (2020) that underscores the power of monocular depth cues. While stereo depth cues play a role in enhancing 3D perception, monocular cues such as motion parallax, shadows, and image blur are often even more effective in VR. Head tracking amplifies these effects, providing users with a more realistic and interpretable view of the data. However, it's noted that these benefits only materialize if the virtual environment is modeled accurately in terms of scale and proportion. In conclusion, the ability to interact with the visualization, such as moving around or rotating, can significantly enhance the understanding of complex 3D visualizations.

Challenges and Best Practices

While VR visualizations allow for an immersive experience, there are notable challenges, such as visual distortion in 2D projections of 3D data. The lecture generally advises against using 3D representations unless the data is inherently 3D. For example, a 3D scatter plot in VR allows users to explore relationships between variables more intuitively than a static 2D representation.

For more conventional applications, 2D visualizations remain the gold standard due to their clarity and precision. The lecture references best practices for selecting encodings in both 2D and 3D contexts, emphasizing the importance of avoiding overcomplicated visual representations.

Conclusion

The lecture provides a comprehensive overview of how visual encodings impact the effectiveness of data visualizations, both in 2D and when extended into 3D and VR environments. By building on foundational research in graphical perception, it outlines best practices for selecting visual encodings based on data type and visualization context. The transition from 2D to 3D or VR visualizations introduces new opportunities for immersive data exploration, but also requires careful consideration to avoid misrepresentation of the data.

References

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. ACM Transactions on Graphics (TOG), 5(2), 110–141. https://doi.org/10.1145/22949.22950

Cleveland, William S., and Robert McGill. "Graphical perception: Theory, experimentation, and application to the development of graphical methods." Journal of the American statistical association 79.387 (1984): 531-554.

LaValle, S. M. (2020). Virtual Reality. Cambridge University Press (expected). http://lavalle.pl/vr/node158.html

https://clauswilke.com/dataviz/no-3d.html

Assignments

This theory lecture has no assignments. The extra time will be used to assist with assignments 1-2.

2.4 Lecture 4: Accessibility in visualization

Methods

Distance learning (Teams lecture, assignment during lecture)

Introduction

Importance of Accessibility in Visualization

Accessibility in data visualization is crucial to ensure that information is comprehensible and usable by all individuals, including those with disabilities. By considering accessibility, we promote inclusivity and enhance the overall communication of information for a broader audience.

Lecture Overview: "Accessibility in Visualization"

In the lecture, we covered the following key topics:

1. People with Disabilities

We explored the different types of disabilities, such as visual impairments, color blindness, and cognitive challenges, and how they affect the way individuals interact with visual data.

2. What is Accessibility?

We defined accessibility and its role in making visual content more inclusive, ensuring that everyone, regardless of ability, can access and understand the data presented.

3. Accessible Visualizations

Key aspects of creating accessible visualizations were discussed, including:

Fonts: Choosing clear and readable fonts.

Colors: Ensuring that color palettes are accessible for especially those with color blindness.

Contrast: Providing sufficient contrast to make elements distinguishable.

Alternative Texts: Writing descriptions for visual elements that can be accessed by screen readers.

Understandability: Making the overall structure of the visualization easy to follow.

Practicality

Concrete Examples of Accessible Visualizations

We reviewed examples that demonstrated how accessible design principles are applied in practice, highlighting what makes a visualization inclusive.

Practice: Writing Alternative Texts

Students practiced crafting alternative texts for various visual elements, ensuring that key information is conveyed even without visual context.

Evaluation of Existing Visualizations

At the end of the lecture, students evaluated the accessibility of an existing visualization and proposed corrections to enhance its accessibility, focusing on areas such as color usage, font clarity, and the addition of alternative texts.

Conclusion

The lecture on "Accessibility in Visualization" emphasized the importance of designing visualizations that are inclusive for all users, particularly those with disabilities. By covering key topics such as the importance of accessibility, effective use of fonts, colors, contrast, alternative texts, and overall understandability, students gained a comprehensive understanding of how to create accessible visual content. Through practical exercises, including writing alternative texts and evaluating existing visualizations, students were able to apply these concepts in real-world scenarios. Ultimately, this lecture highlighted that making visualizations accessible not only fosters inclusivity but also enhances clarity and usability for everyone.

References

WebAIM: Contrast Checker
Accessible fonts: How to choose a font for web accessibility (siteimprove.com)
WHO: Disability
Everything you need to know to write effective alt text - Microsoft Support
ACC.320: Accessible Information Visualization | Tampere universities (tuni.fi)

Met opmerkingen [JM2]: @Paullina Baltzar (TAU) Your lecture on accessible visualizations